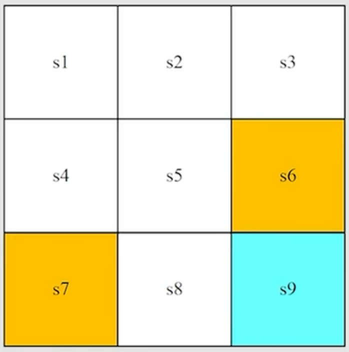
Lecture 1: Basic Concepts

1.State:

Agent相对于环境的状态，包含位置，速度等因素

- For the grid-world example, the location of the agent is the state. There are nine possible locations and hence nine states: $s_1, s_2, \ldots, s_9$.



2.State Space:

所有状态的集合 $\mathcal{S} = \{s_i\}_{i=1}^{9}$

3.Action:

在每一个状态，可采取的一系列行动

- a_1: move upwards;
- a_2: move rightwards;
- a_3: move downwards;
- a_4: move leftwards;
- a_5: stay unchanged;

4.Action Space of a state:

在一个状态下所有可以采取的行动集合$\mathcal{A}(s_i) = \{a_i\}_{i=1}^{5}$

5.State transition:

当采取一个action，agent从一个状态转移到另一个状态。$s_1 \xrightarrow{a_2} s_2$

6.Forbidden area:

√case1:可以进入但是会受到惩罚

case2:不可以进入，返回到原状态

7.Tabular represention:

表格形式的表达，但只能针对确定性的情况，即已知所处状态采取的Action

	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	a_5 (unchanged)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

8.State transition probability:

一般地，对于更复杂的情况（action是不确定的），我们使用条件概率来描述State transition

- Intuition: At state s_1, if we choose action a_2, the next state is s_2.
- Math:

$$p(s_2|s_1, a_2) = 1$$
$$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$$

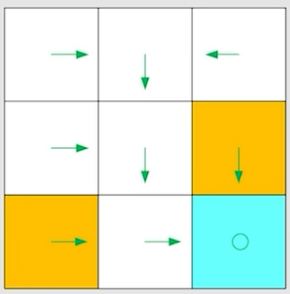
9.Policy:π

告诉Agent在某个state应该采取什么样的action

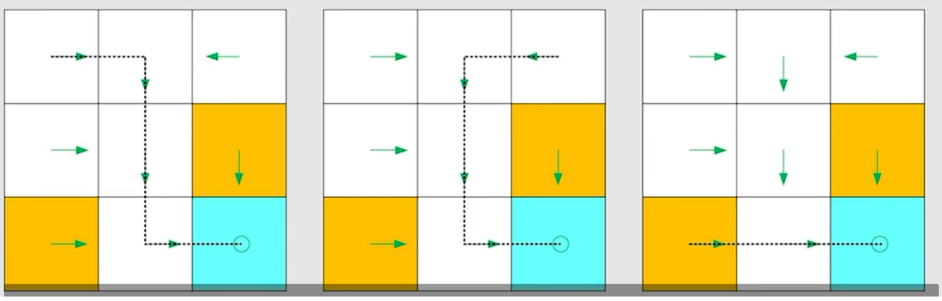
表示方式：

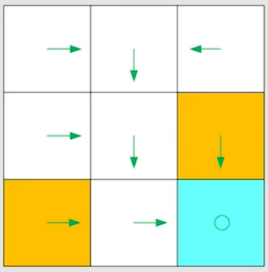
- 确定性的策略

Intuitive representation: The arrows demonstrate a policy.



Based on this policy, we get the following paths with different starting points.





Mathematical representation: using conditional probability

For example, for state s_1:

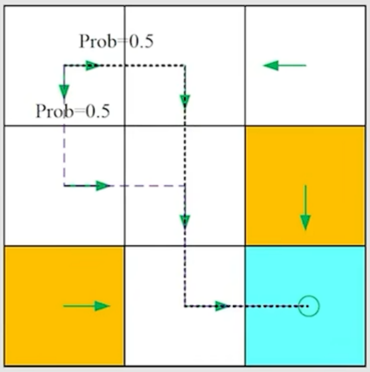
$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 1$$
$$\pi(a_3|s_1) = 0$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$

It is a deterministic policy.

- 随机策略

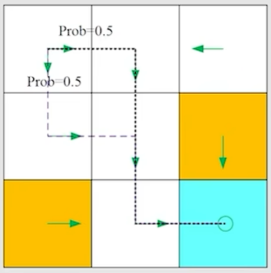
There are stochastic policies.

For example:



In this policy, for s_1:

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 0.5$$
$$\pi(a_3|s_1) = 0.5$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$



Tabular representation of a policy: how to use this table.

	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	a_5 (unchanged)
s_1	0	0.5	0.5	0	0
s_2	0	0	1	0	0
s_3	0	0	0	1	0
s_4	0	1	0	0	0
s_5	0	0	1	0	0
s_6	0	0	1	0	0
s_7	0	1	0	0	0
s_8	0	1	0	0	0
s_9	0	0	0	0	1

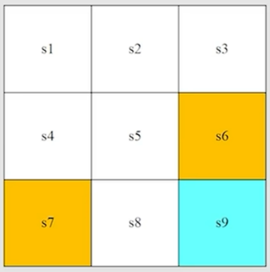
Can represent either *deterministic* or *stochastic* cases.

10.Reward:

RL中的概念，在agent采取一个action后得到的一个实数。人机交互的一种方式。

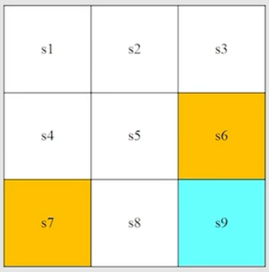
如果是正数，代表该action是被鼓励的，反之，则不鼓励。

reward = 0，代表不惩罚



In the grid-world example, the rewards are designed as follows:

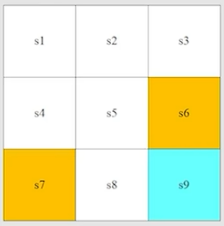
- If the agent attempts to get out of the boundary, let $r_{\text{bound}} = -1$
- If the agent attempts to enter a forbidden cell, let $r_{\text{forbid}} = -1$
- If the agent reaches the target cell, let $r_{\text{target}} = +1$
- Otherwise, the agent gets a reward of $r = 0$.



Tabular representation of *reward transition*: how to use the table?

	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	a_5 (unchanged)
s_1	r_{bound}	0	0	r_{bound}	0
s_2	r_{bound}	0	0	0	0
s_3	r_{bound}	r_{bound}	r_{forbid}	0	0
s_4	0	0	r_{forbid}	r_{bound}	0
s_5	0	r_{forbid}	0	0	0
s_6	0	r_{bound}	r_{target}	0	r_{forbid}
s_7	0	0	r_{bound}	r_{bound}	r_{forbid}
s_8	0	r_{target}	r_{bound}	r_{forbid}	0
s_9	r_{forbid}	r_{bound}	r_{bound}	0	r_{target}

Can only represent *deterministic* cases.



Mathematical description: conditional probability

- Intuition: At state s_1, if we choose action a_1, the reward is -1.
- Math: $p(r=-1|s_1,a_1)=1$ and $p(r\neq -1|s_1,a_1)=0$

Remarks:

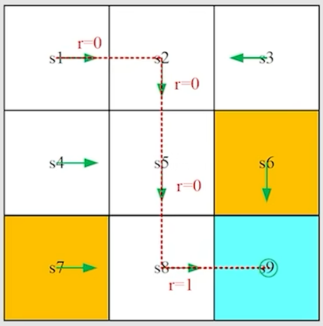
- Here it is a deterministic case. The reward transition could be stochastic.
- For example, if you study hard, you will get rewards. But how much is uncertain.
- The reward depends on the state and action, but not the next state (for example, consider s_1,a_1 and s_1,a_5).

依赖于当前的state和action，与下一个state无关

11.Trajectory and return

trajectory是一个 state-action-reward 链

return 是一个trajectory中所有reward之和



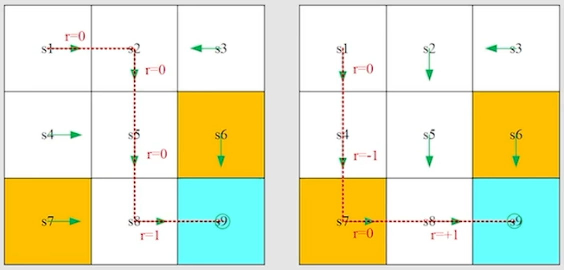
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

return:可以用来评估policy

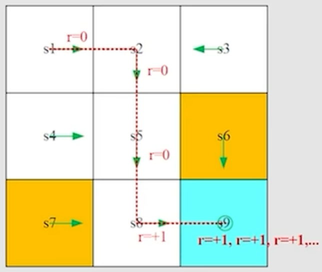


Which policy is better?

- **Intuition**: the first is better, because it avoids the forbidden areas.
- **Mathematics**: the first one is better, since it has a greater return!
- Return could be used to evaluate whether a policy is good or not (see details in the next lecture)!

12.Discounted return

有时，一个trajectory可能是无限的，这时return也是∞，



A trajectory may be infinite:

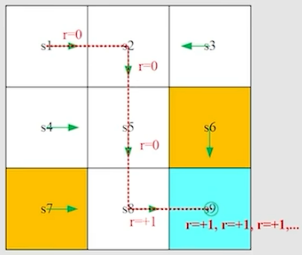
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \dots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \cdots = \infty$$

The definition is invalid since the return diverges!

此时，通过引入折扣因子 γ



Need to introduce a *discount rate* $\gamma \in [0, 1)$
Discounted return:

$$\begin{aligned} \text{discounted return} &= 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \dots \\ &= \gamma^3 (1 + \gamma + \gamma^2 + \dots) = \gamma^3 \frac{1}{1-\gamma}. \end{aligned}$$

Roles: 1) the sum becomes finite; 2) balance the far and near future rewards:

- If γ is close to 0, the value of the discounted return is dominated by the rewards obtained in the near future.
- If γ is close to 1, the value of the discounted return is dominated by the rewards obtained in the far future.

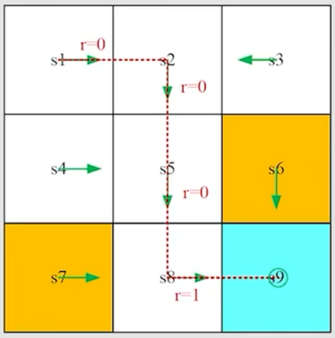
13.Episode

在一个trajectory中，如果agent在terminal states中停止，就称该trajectory为一个episode

所以说，episode是有限步的trajectory，他的一个特殊情况。

这样的episode的任务也成为episodic tasks 间歇性的任务。

When interacting with the environment following a policy, the agent may stop at some *terminal states*. The resulting trajectory is called an *episode* (or a trial).



Example: episode

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

An episode is usually assumed to be a finite trajectory. Tasks with episodes are called *episodic tasks*.

那么对于不存在terminal stases的continuing tasks，又该如何处理。（现实中不存在）

我们可以将episodic tasks转换为continuing tasks

option 1:当在target state时，让agent一直处于该状态，不让其离开，并且设置之后的reward=0

√option 2:正常对待target state，设置reward = +1，可以跳出该state，和正常state一样。

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

14.Markov decision process (MDP)

Markov process中的policy是确定的

MDP中的要素

- Sets:集合
  - State: 状态$\mathcal{S}$的集合

    - Action:动作 $\mathcal{A}(s)$的集合
    - Reward:奖励 $\mathcal{R}(s,a)$的集合
- Probability distribution:概率分布
    - State transition probability:状态转移概率

      在状态 s 下采取动作 a 到达状态 s' 的概率 $p\left(s'|s,a\right)$
    - Reward probability:奖励概率

      在状态 s 下采取动作 a 得到奖励 r 的概率 $p\left(r|s,a\right)$
- Policy: 策略

  在状态 s ，采取动作 a 的概率 $\pi(a|s)$
- Markov property: 无记忆性

  $p(s_{t+1}|a_{t+1},s_t,\ldots,a_1,s_0)=p(s_{t+1}|a_{t+1},s_t),$

  $p(r_{t+1}|a_{t+1},s_t,\ldots,a_1,s_0)=p(r_{t+1}|a_{t+1},s_t).$

无论是 s_{t+1} 还是 r_{t+1} 都与之前的状态和动作无关，只与上一步的有关。